

WHAT IS CLAIMED IS:

- 1 1. A method, comprising:
 - 2 transferring from a first non-volatile storage unit of a plurality of non-volatile
 - 3 storage units, to a logic engine of a storage processor having a cache memory, a first unit
 - 4 of data stored in a stripe across said plurality of non-volatile storage units, in a first
 - 5 transfer operation which bypasses said cache memory;
 - 6 transferring from a second non-volatile storage unit of said plurality of non-
 - 7 volatile storage units, to said logic engine, a second unit of data stored in said stripe, in a
 - 8 second transfer operation which bypasses said cache memory; and
 - 9 constructing in said logic engine a third unit of data using said first unit of data
 - 10 transferred to said logic engine in said first transfer operation and using said second unit
 - 11 of data transferred to said logic engine in said second transfer operation.
- 1 2. The method of claim 1 further comprising transferring said constructed
- 2 third unit of data to a third non-volatile storage unit of said plurality of non-volatile
- 3 storage units in a third transfer operation, and storing said constructed third unit of data in
- 4 said stripe.
- 1 3. The method of claim 2 wherein said third transfer operation bypasses said
- 2 cache memory.
- 1 4. The method of claim 2 wherein said plurality of non-volatile storage units
- 2 are arranged in a Redundant Array of Independent Disks organization.

1 5. The method of claim 4 wherein said second unit of data is parity data.

1 6. The method of claim 5 wherein each non-volatile storage unit is a disk
2 drive.

1

2 7. The method of claim 6 wherein said stripe is organized as a Redundant
3 Array of Independent Disks stripe of blocks of data including a block of parity data and
4 each of said units of data is a block of data of said Redundant Array of Independent Disks
5 stripe.

1 8. The method of claim 7 wherein said logic engine includes a store queue
2 capable of storing a block of data from said Redundant Array of Independent Disks
3 stripe.

1 9. The method of claim 1 wherein said cache memory of said logic engine is
2 a random access memory.

1 10. The method of claim 1 wherein said constructing in said logic engine a third
2 unit of data includes applying said first unit of data transferred to said logic engine in said
3 first transfer operation, to a store queue of said logic engine and performing an exclusive-
4 OR logic function operation on the contents of said first store queue and said applied first
5 unit of data, and includes applying said second unit of data transferred to said logic
6 engine in said second transfer operation, to said store queue of said logic engine and

7 performing an exclusive-OR logic function operation on the contents of said store queue
8 and said applied second unit of data.

1 11. The method of claim 2 further comprising:

2 transferring from a fourth non-volatile storage unit of said plurality of non-
3 volatile storage units, to said logic engine, a fourth unit of data stored in said stripe, in a
4 fourth transfer operation which bypasses said cache memory; and

5 transferring from a fifth non-volatile storage unit of said plurality of non-volatile
6 storage units, to said logic engine, a fifth unit of data stored in said stripe, in a fifth
7 transfer operation which bypasses said cache memory; and

8 wherein said constructing in said logic engine said third unit of data also uses said
9 fourth unit of data transferred to said logic engine in said fourth transfer operation and
10 using said fifth unit of data transferred to said logic engine in said fifth transfer operation
11 prior to said transferring said constructed third unit of data to said third non-volatile
12 storage unit in said third transfer operation.

1 12. The method of claim 2 further comprising:

2 for each additional non-volatile storage unit of said plurality of storage units:

3 transferring from an additional non-volatile storage unit of said plurality of
4 non-volatile storage units, to said logic engine, an additional unit of data stored in said
5 stripe, in an additional transfer operation which bypasses said cache memory; and

6 wherein said constructing in said logic engine said third unit of data also uses said
7 additional unit of data transferred to said logic engine in said additional transfer operation
8 prior to said transferring said constructed third unit of data to said third non-volatile
9 storage unit in said third transfer operation.

1 13. The method of claim 1 further comprising:

2 transferring to said logic engine, a fourth unit of data which is new data to be
3 stored in said stripe; wherein said first unit of data is old data to be replaced by said new
4 data in said stripe, said second unit of data is old parity data to be replaced in said stripe
5 and wherein said constructing in said logic engine said third unit of data constructs a
6 replacement unit of parity data using said new data transferred to said logic engine, said
7 method further comprising transferring said third unit of new parity data to said second
8 non-volatile storage unit to replace said second unit of old parity data.

1 14. The method of claim 1 further comprising transferring said constructed
2 third unit of data to a host in a third transfer operation..

1 15. The method of claim 1 wherein the storage processor issues read
2 commands to said first and second non-volatile storage units so that said first and second
3 units of data are accessed from said first and second non-volatile storage units at least
4 partially in parallel.

1 16. An article comprising a storage medium, the storage medium comprising
2 machine readable instructions stored thereon to:

3 transfer from a first non-volatile storage unit of a plurality of non-volatile storage
4 units, to a logic engine of a storage processor having a cache memory, a first unit of data
5 stored in a stripe across said plurality of non-volatile storage units, in a first transfer
6 operation which bypasses said cache memory;

7 transfer from a second non-volatile storage unit of said plurality of non-volatile
8 storage units, to said logic engine, a second unit of data stored in said stripe, in a second
9 transfer operation which bypasses said cache memory; and

10 construct in said logic engine a third unit of data using said first unit of data
11 transferred to said logic engine in said first transfer operation and using said second unit
12 of data transferred to said logic engine in said second transfer operation.

1 17. The article of claim 16 wherein the storage medium further comprises
2 machine readable instructions stored thereon to transfer said constructed third unit of data
3 to a third non-volatile storage unit of said plurality of non-volatile storage units in a third
4 transfer operation, and store said constructed third unit of data in said stripe.

1 18. The article of claim 17 wherein said third transfer operation bypasses said
2 cache memory.

1 19. The article of claim 17 wherein said plurality of non-volatile storage units
2 are arranged in a Redundant Array of Independent Disks organization.

1 20. The article of claim 19 wherein said second unit of data is parity data.

1 21. The article of claim 20 wherein each non-volatile storage unit is a disk
2 drive.

1 22. The article of claim 21 wherein said stripe is organized as a Redundant
2 Array of Independent Disks stripe of blocks of data including a block of parity data and
3 each of said units of data is a block of data of said Redundant Array of Independent Disks
4 stripe.

1 23. The article of claim 22 wherein said logic engine includes a store queue
2 capable of storing a block of data from said Redundant Array of Independent Disks
3 stripe.

1 24. The article of claim 16 wherein said cache memory of said logic engine is
2 a random access memory.

1 25. The article of claim 16 wherein the machine readable instructions to construct
2 in said logic engine a third unit of data include machine readable instructions stored on
3 the storage medium to apply said first unit of data transferred to said logic engine in said
4 first transfer operation, to a store queue of said logic engine and perform an exclusive-OR
5 logic function operation on the contents of said first store queue and said applied first unit
6 of data, and to apply said second unit of data transferred to said logic engine in said
7 second transfer operation, to said store queue of said logic engine and perform an
8 exclusive-OR logic function operation on the contents of said store queue and said
9 applied second unit of data.

1 26. The article of claim 17 wherein the storage medium further comprises
2 machine readable instructions stored thereon to:

3 transfer from a fourth non-volatile storage unit of said plurality of non-volatile
4 storage units, to said logic engine, a fourth unit of data stored in said stripe, in a fourth
5 transfer operation which bypasses said cache memory; and

6 transfer from a fifth non-volatile storage unit of said plurality of non-volatile
7 storage units, to said logic engine, a fifth unit of data stored in said stripe, in a fifth
8 transfer operation which bypasses said cache memory; and

9 wherein the machine readable instructions to construct in said logic engine a third

10 unit of data include machine readable instructions stored on the storage medium to use
11 said fourth unit of data transferred to said logic engine in said fourth transfer operation
12 and to use said fifth unit of data transferred to said logic engine in said fifth transfer
13 operation prior to said transferring said constructed third unit of data to said third non-
14 volatile storage unit in said third transfer operation.

1 27. The article of claim 16 wherein the storage medium further comprises
2 machine readable instructions stored thereon to:

3 transfer to said logic engine, a fourth unit of data which is new data to be stored in
4 said stripe; wherein said first unit of data is old data to be replaced by said new data in
5 said stripe, said second unit of data is old parity data to be replaced in said stripe and
6 wherein the machine readable instructions to construct in said logic engine a third unit of
7 data include machine readable instructions stored on the storage medium to construct a
8 replacement unit of parity data using said new data transferred to said logic engine, said
9 wherein the storage medium further comprises machine readable instructions stored
10 thereon to transfer said third unit of new parity data to said second non-volatile storage
11 unit to replace said second unit of old parity data.

1 28. The article of claim 16 wherein the storage medium further comprises
2 machine readable instructions stored thereon to transfer said constructed third unit of data
3 to a host in a third transfer operation.

1 29. The article of claim 16 wherein the storage medium further comprises
2 machine readable instructions stored thereon to issue read commands to said first and
3 second non-volatile storage units so that said first and second units of data are accessed
4 from said first and second non-volatile storage units at least partially in parallel.

1 30. A system, comprising:
2 at least one memory which includes an operating system and an
3 application;
4 a processor coupled to the memory;
5 data storage having a plurality of non-volatile storage units;
6 a data storage processor adapted to manage Input/Output (I/O) access to
7 the data storage and having a cache memory and a logic engine; and
8 a device driver executable by the processor in the memory, wherein at
9 least one of the application, operating system, device driver and the storage processor is
10 adapted to:
11 transfer from a first non-volatile storage unit to said logic engine of said storage
12 processor, a first unit of data stored in a stripe across said plurality of non-volatile storage
13 units, in a first transfer operation which bypasses said cache memory;
14 transfer from a second non-volatile storage unit of said plurality of non-volatile
15 storage units, to said logic engine, a second unit of data stored in said stripe, in a second
16 transfer operation which bypasses said cache memory; and
17 construct in said logic engine a third unit of data using said first unit of data
18 transferred to said logic engine in said first transfer operation and using said second unit
19 of data transferred to said logic engine in said second transfer operation.

1 31. The system of claim 30 wherein said at least one of the application,
2 operating system, device driver and the storage processor is further adapted to transfer
3 said constructed third unit of data to a third non-volatile storage unit of said plurality of
4 non-volatile storage units in a third transfer operation, and store said constructed third
5 unit of data in said stripe.

1 32. The system of claim 31 wherein said third transfer operation bypasses said
2 cache memory.

1 33. The system of claim 31 wherein said plurality of non-volatile storage units
2 are arranged in a Redundant Array of Independent Disks organization.

1 34. The system of claim 33 wherein said second unit of data is parity data.

1 35. The system of claim 34 wherein each non-volatile storage unit is a disk
2 drive.

1 36. The system of claim 35 wherein said stripe is organized as a Redundant
2 Array of Independent Disks stripe of blocks of data including a block of parity data and
3 each of said units of data is a block of data of said Redundant Array of Independent Disks
4 stripe.

1 37. The system of claim 36 wherein said logic engine includes a store queue
2 capable of storing a block of data from said Redundant Array of Independent Disks
3 stripe.

1 38. The system of claim 30 wherein said cache memory of said logic engine is
2 a random access memory.

1 39. The system of claim 30 wherein to construct in said logic engine a third unit
2 of data, said at least one of the application, operating system, device driver and the

3 storage processor is further adapted to apply said first unit of data transferred to said logic
4 engine in said first transfer operation, to a store queue of said logic engine and perform
5 an exclusive-OR logic function operation on the contents of said first store queue and
6 said applied first unit of data, and to apply said second unit of data transferred to said
7 logic engine in said second transfer operation, to said store queue of said logic engine and
8 perform an exclusive-OR logic function operation on the contents of said store queue and
9 said applied second unit of data.

1 40. The system of claim 31 wherein said at least one of the application, operating
2 system, device driver and the storage processor is further adapted to transfer from a
3 fourth non-volatile storage unit of said plurality of non-volatile storage units, to said logic
4 engine, a fourth unit of data stored in said stripe, in a fourth transfer operation which
5 bypasses said cache memory; and

6 transfer from a fifth non-volatile storage unit of said plurality of non-volatile
7 storage units, to said logic engine, a fifth unit of data stored in said stripe, in a fifth
8 transfer operation which bypasses said cache memory; and

9 wherein to construct in said logic engine a third unit of data, said at least one of
10 the application, operating system, device driver and the storage processor is further
11 adapted to use said fourth unit of data transferred to said logic engine in said fourth
12 transfer operation and to use said fifth unit of data transferred to said logic engine in said
13 fifth transfer operation prior to said transferring said constructed third unit of data to said
14 third non-volatile storage unit in said third transfer operation.

1 41. The system of claim 30 wherein said at least one of the application, operating
2 system, device driver and the storage processor is further adapted to transfer to said logic
3 engine, a fourth unit of data which is new data to be stored in said stripe; wherein said
4 first unit of data is old data to be replaced by said new data in said stripe, said second unit
5 of data is old parity data to be replaced in said stripe and wherein the machine readable

6 instructions to construct in said logic engine a third unit of data include machine readable
7 instructions stored on the storage medium to construct a replacement unit of parity data
8 using said new data transferred to said logic engine, said wherein the storage medium
9 further comprises machine readable instructions stored thereon to transfer said third unit
10 of new parity data to said second non-volatile storage unit to replace said second unit of
11 old parity data.

1 42. The system of claim 30 wherein said at least one of the application,
2 operating system, device driver and the storage processor is further adapted to transfer
3 said constructed third unit of data to a host in a third transfer operation.

1

2 43. The system of claim 30 wherein said at least one of the application,
3 operating system, device driver and the storage processor is further adapted to issue read
4 commands to said first and second non-volatile storage units so that said first and second
5 units of data are accessed from said first and second non-volatile storage units at least
6 partially in parallel.

1 44. A device for use with a data storage having a plurality of non-volatile
2 storage units, comprising:

3 a data storage processor adapted to manage Input/Output (I/O) access to
4 the data storage and having a cache memory and a logic engine, wherein the storage
5 processor is further adapted to:

6 transfer from a first non-volatile storage unit to said logic engine of said
7 storage processor, a first unit of data stored in a stripe across said plurality of non-volatile
8 storage units, in a first transfer operation which bypasses said cache memory;

9 transfer from a second non-volatile storage unit of said plurality of non-
10 volatile storage units, to said logic engine, a second unit of data stored in said stripe, in a
11 second transfer operation which bypasses said cache memory; and

12 construct in said logic engine a third unit of data using said first unit of
13 data transferred to said logic engine in said first transfer operation and using said second
14 unit of data transferred to said logic engine in said second transfer operation.

1 45. The device of claim 44 wherein the device is further adapted to transfer
2 said constructed third unit of data to a third non-volatile storage unit of said plurality of
3 non-volatile storage units in a third transfer operation, and store said constructed third
4 unit of data in said stripe.

1 46. The device of claim 45 wherein said third transfer operation bypasses said
2 cache memory.

1 47. The device of claim 45 wherein each non-volatile storage unit is a
2 disk drive, said plurality of non-volatile storage units are arranged in a Redundant Array
3 of Independent Disks organization, said stripe is organized as a Redundant Array of
4 Independent Disks stripe of blocks of data including a block of parity data and each of
5 said units of data is a block of data of said Redundant Array of Independent Disks stripe.

1 48. The device of claim 47 wherein said logic engine includes a store queue
2 capable of storing a block of data from said Redundant Array of Independent Disks
3 stripe, said cache memory of said logic engine is a random access memory, and to
4 construct in said logic engine a third unit of data, said storage processor is further adapted
5 to apply said first unit of data transferred to said logic engine in said first transfer
6 operation, to a store queue of said logic engine and perform an exclusive-OR logic

7 function operation on the contents of said first store queue and said applied first unit of
8 data, and to apply said second unit of data transferred to said logic engine in said second
9 transfer operation, to said store queue of said logic engine and perform an exclusive-OR
10 logic function operation on the contents of said store queue and said applied second unit
11 of data.

1 49. The device of claim 44 wherein said storage processor is further adapted
2 to transfer said constructed third unit of data to a host in a third transfer operation.

1 50. The device of claim 44 wherein said storage processor is further adapted
2 to issue read commands to said first and second non-volatile storage units so that said
3 first and second units of data are accessed from said first and second non-volatile storage
4 units at least partially in parallel.